Digital Image Processing Exercise Report

Instructor: MITSUGAMI IKUHISA

Student ID: 81322400098

Student Name: GARCIA AARON CALVIN

Date: 2024 05 28

**Summary/Discussion:**

Hough-Line Transformation

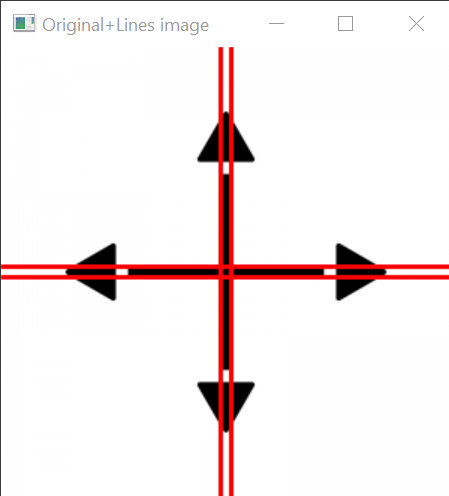
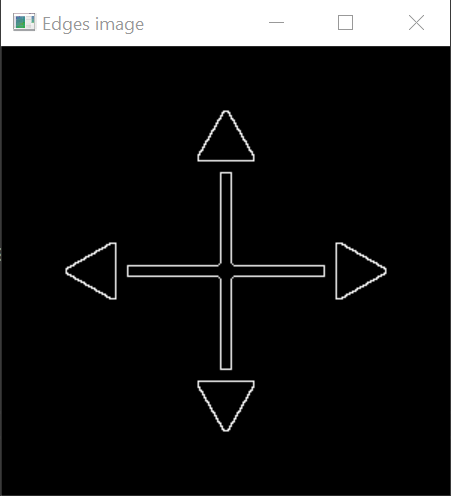
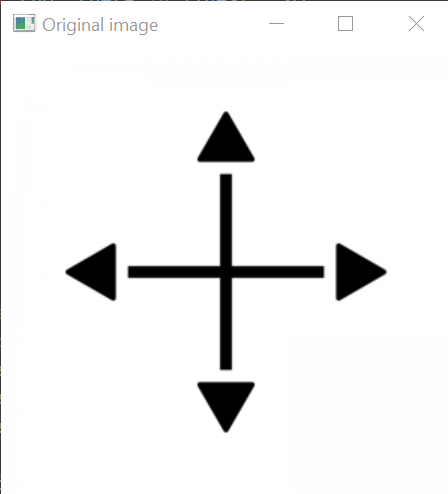
The Hough-Line Transform is an algorithm whose main functionality is to detect straight lines within an image. It is mainly used in computer vision applications and image processing useful in identifying lines in edge-detected images, where edges represent significant transitions in pixel intensity. My implementation begins by utilizing the Canny-edge detection function in cv2, copied from my last report, where I used a wrapper funciton to make it easier to keep track of the image/object I would be applying the Hough-Line transformation algorithm on. After the detecting edges, I used cv2’s Hough-Line Transformation function to identify lines the image I generated in the image. The function then returns a mapped “image” value of all the lines it detected in the image. This is the all of the lines that were calculated in the generated Parameter (Hough) Space, which are created, in short, by calculating ρ=xcosθ+ysinθ, where x and y are points on the original image graph.

As we can see, the result were able to generate 8 solid lines, however, I expected there to be more considering the diagonals of the arrows.

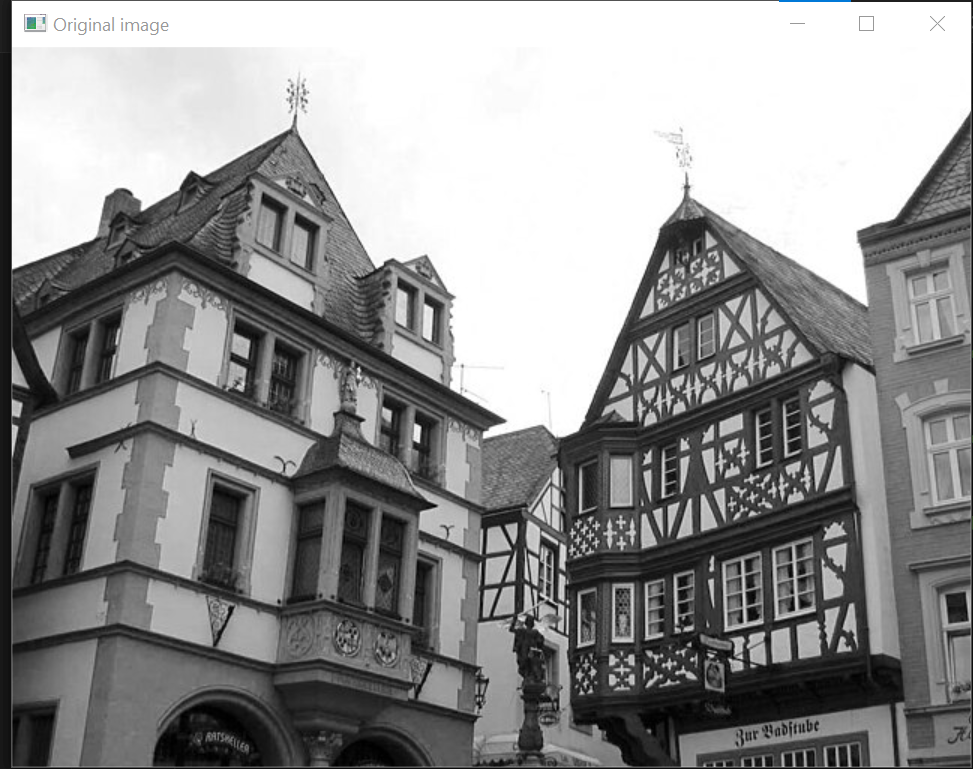
Finding Global Intesity Value Threshold and using Otsu’s Method

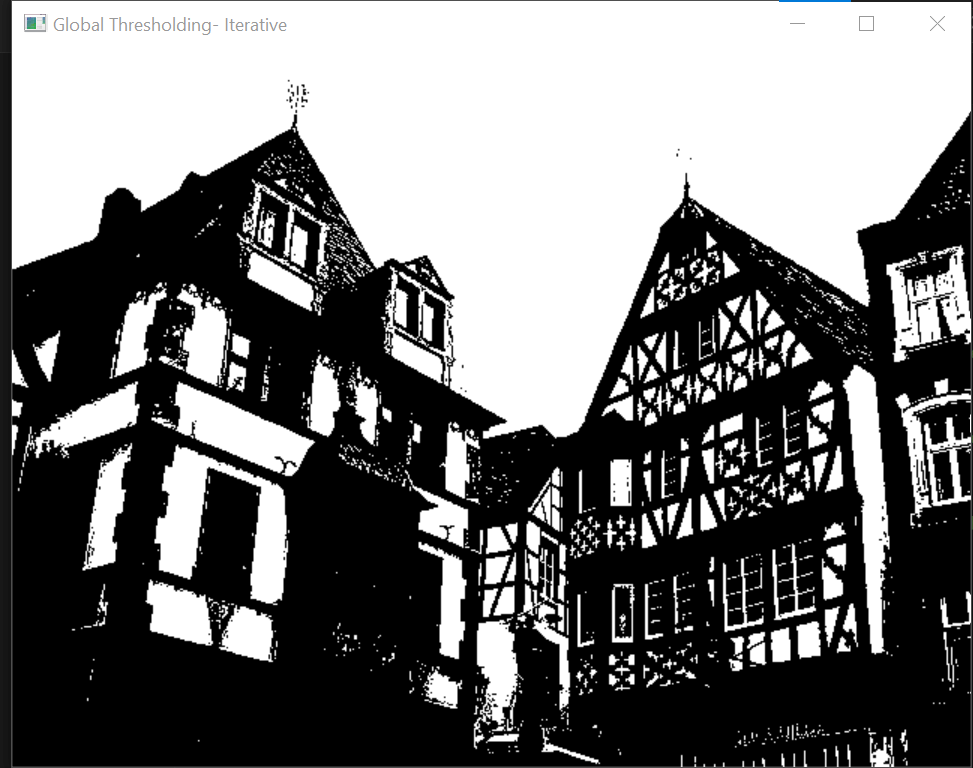
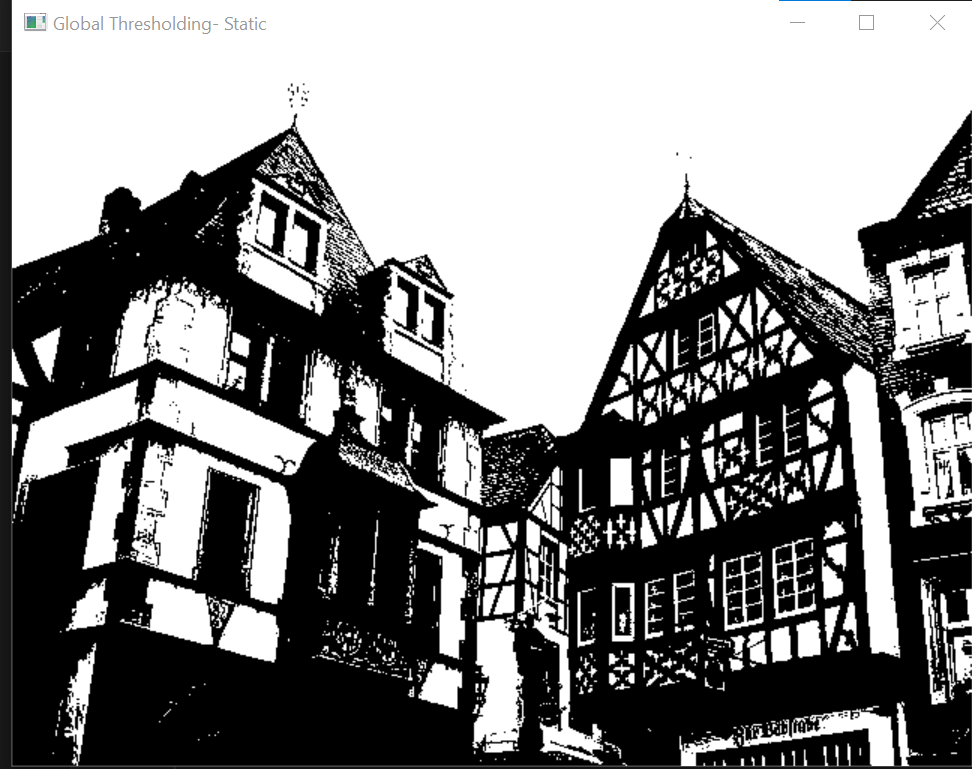
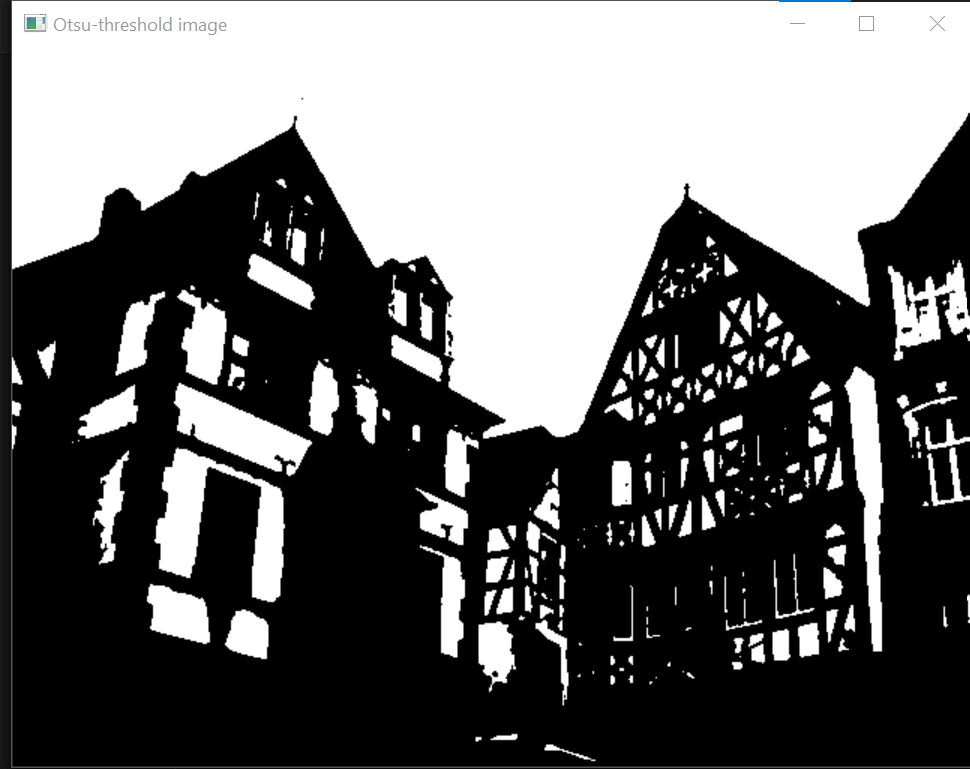
Within my code, I implemented Global Threshold-ing twice, once with a static threshold value of 127 and the second time using an algorithm given to me by the slides in order to look into the image iteratively and find the threshold-ing value of best fit. To do this, I looked into the images intensity values, summed and divided them into 2 medians- one on each side of the intial thresholding guess: 127. After that I divided it by 2. I repeat all of the above until the rate of change in the threshold value compared to its previous value equaled zero, after which I applied thresholding to the image (that value was about ~150). Lastly, I used cv2’s implementation of Otsu's Method to determine an optimal threshold. However before that, I applied Gaussian blurring to the image to reduce noise and enhance the quality of the threshold-ing that Otsu’s Method has. You can see the improvements in the binary segmentation with each output.

**Hough-Line Transformation Output:**



Global Static/Iterative Thresholding/Otsu’s Method Outputs



**References**

1. Week 8 Slides